Collaborative Mapping and Search for Autonomous Helicopters

Eric N. Johnson
eric.johnson@gatech.edu
Lockheed Martin
Professor of Avionics

John G. Mooney
john.g.mooney@gatech.edu
Graduate Research Assistant

Daniel P. Magree
dmagree3@gatech.edu
Graduate Research Assistant

Georgia Institute of Technology
Atlanta, Georgia

Matthew White
matthew.white@sikorsky.com
Engineer,
Handling Qualities
and Control Laws

Jonathan Hartman
jonathan.hartman@sikorsky.com
Engineer,
Advanced Concepts

Vineet Sahasrabudhe
vsahasrabudhe@sikorsky.com
Manager,
Handling Qualities
and Control Laws

Sikorsky Aircraft Corporation
Stratford, Connecticut

ABSTRACT

This paper describes recent results from a partnership between the Sikorsky Aircraft Corporation and the Georgia Institute of Technology to develop, improve, and flight test a multi-aircraft collaborative architecture, focused on decentralized autonomous decision-making. The architecture includes a finite-state machine, Voronoi mapping strategy, and real-time information sharing system designed to solve a challenge problem. The architecture was implemented on a pair of Yamaha RMax helicopters outfitted with modular avionics, as well as an associated set of simulation tools. Simulation results for single- and multiple-aircraft scenarios are presented, along with a quadratic relationship between mapping speed and task completion time. Further work suggested includes validation of simulation results in flight test with two real aircraft, as well as further exploration between search problem parameters and theoretical optimal performance.

INTRODUCTION

The market for unmanned aerial vehicles continues to grow. In addition to the well-publicized military uses of unmanned and remotely piloted aircraft, there is an increased recognition of legitimate civil applications, such as law enforcement, search and rescue, pipeline and power line inspections, and so forth.

In light of the ever-expanding applications and requirements, using a collaborating team, or swarm, of UASs will have many advantages over operating a single UAS. Owners/operators can invest in a number of simple, inexpensive aircraft, rather than a single aircraft. A swarm presents a high degree of robustness, as the loss of a single aircraft no longer represents mission failure, nor excessive cost of replacement. Moreover, the damage induced by a crash to people or property on the ground becomes limited. By definition, a swarm is in many places at once, meaning a wider sensor net can be cast. Heterogeneous sensor packages, which can be tailored to the mission and environment, can be added to or removed from the swarm very easily.

However, there are costs associated with UAS collaboration. Establishing communication, task allocation, coordination, synchronization, collision avoidance, and an effective user interface are just part of the problem.

This paper focuses on use of one or two heterogeneous aircraft to collaborate to solve a scenario-driven challenge problem. The aircraft are given the mission to find a fugitive who has entered a small urban area, where he has confederates ready to defend him against arrest. The aircraft need to collaboratively and rapidly scan/map the area from a low altitude, find the fugitive's hiding place, and force him into the open. Additionally, this paper presents a monocular vision-based mapping system and its integration with collaborative map and search.

Presented at the AHS 69th Annual Forum, Phoenix, Arizona, May 21–23, 2013.

Multi-robot coordination and information sharing has been studied by many researchers, primarily in the last 20 years. These efforts have investigated multiple paradigms of control and coordination. Arkin (Ref. 1), Balch (Ref. 2), and Parker (Ref. 3), for example, have focused on reactive behavior-based control and interaction. This approach assumes minimal or no direct communication between robots, often relying upon the robot's ability to observe the behavior of other robots to coordinate efforts. Other efforts have maintained decentralized control, but allowed robots to explicitly share state information (Ref. 4). Others use a fully-centralized approach, treating the system as a single "meta-robot" with a very high-dimension configuration space (Refs. 5 and 6). Coordination of robotic aircraft has been studied extensively as well, though usually in the context of collision avoidance or formation control (Refs. 7 through 9). Previous work on autonomous collaborative search has demonstrated the effectiveness of spiral and lane-based search patterns using appropriate objective functions and heuristics (Refs. 10 and 11). Vision-based mapping for UAVs has been investigated in the context of simultaneous localization and mapping in Weiss et al. (Ref. 12) and in Chawdhary et al. (Ref. 13).

One note about terminology: in discussing the autonomous collaborative decision-making of multiple aircraft, the language can quickly because confusing and ambiguous with regards to which aircraft is being referenced. For the purposes of this paper, when referring to each aircraft's own resources, information, or decisions, the term "own ship" will be used and any aircraft outside of own-ship will be called "other-ship."

GENERAL APPROACH

The challenge problem described in the INTRODUCTION includes many extremely complex aspects and was approached in a methodical manner to optimize key technology development. In order to limit the scope of this work, several assumptions were made to establish a baseline approach. These assumptions may be addressed in later work. Specifically, it is assumed that the suspect is hiding in a building, and that the suspect emerges from hiding and surrenders immediately upon one of the aircraft coming within a specified radius.

State Machine Architecture

For this mission, the aircraft acts as a finite-state machine to transition between relevant flight modes. In MAP mode, the aircraft scans the search area to build terrain model. FIND BLDGS mode analyzes the terrain model to find likely hiding places. SELECT NEXT and SEARCH modes alternate between searching a likely hiding place and selecting which place to search next. If the aircraft finds the fugitive, it transitions into CORRAL mode and otherwise transitions to LOITER if all other tasks have been accomplished.

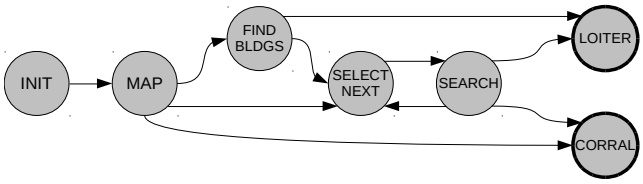

Figure 1. The finite-state machine model of the map and search algorithm.

Terrain Representation

Prior work (Refs. 14 and 15) used three-dimensional evidence grids to maintain a model of the local terrain. This approach was used successfully for obstacle avoidance and had previously been used by Scherer, et al. in a similar manner (Ref. 16). A brief summary of the approach is included here.

An evidence grid represents a space of interest as a three dimensional array, using it to store a measure of likelihood of a particular point in space being occupied. This likelihood is updated when the sensor receives a measurement, according to rules specific to the sensor's characteristics. A particularly useful version of this evidence grid uses log-odds as the measure of likelihood or 'belief':

$$b(o) = \ln\left(\frac{P(o|M)}{1 - P(o|M)}\right)$$

If subsequent measurements are assumed to be independent, then this belief function can be updated by simply adding (or subtracting) the log-odds of occupancy according to the sensor model and new measurements.

$$b_{new}(o) = b_{old}(o) + \ln\left(\frac{P(M_{new}|o)}{1 - P(M_{new}|o)}\right)$$

The last term on the right hand side of this equation is a characteristic of specific sensor chosen. The sensors used in this work include a scanning laser and monocular camera feature point detection.

Subsequent work modified this approach to use a "two-and-a-half-dimensional" paradigm by flattening the three dimensional array into three two-dimensional arrays: one with terrain height, one with a measure of the confidence in that terrain height, and a third keeping track of the time of observation. While the rules to update the 2½ D grids are a bit more complicated than an ordinary evidence map, and there is some information lost, the more compact terrain representation is more conducive for data sharing over a wireless network.

Mapping Strategy

While mapping, the aircraft divides the search area into two categories: observed and unobserved. The algorithm then expands the edges of the observed areas into the unobserved areas until the edges converge on one another. This convergence effectively forms a Voronoi graph. The nodes of this graph represent central points of the unobserved areas. Each node has an associated value which measures the distance of these centers from the nearest edge in a 1-norm sense. Thus, the value of the node roughly indicates the size of the unobserved mass (Figures 2, 3 and 4).

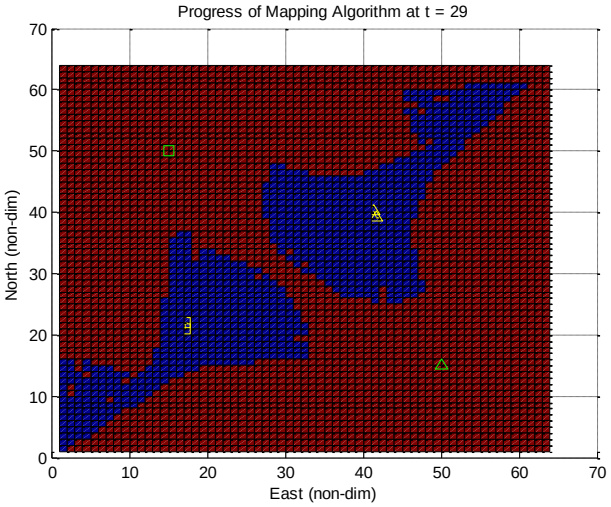

Figure 2. An example of the mapping in progress. The yellow triangles represent the first aircraft seeking the point marked with the green triangle. The yellow squares are the second aircraft seeking the green square. Blue areas have been explored, red areas have not.

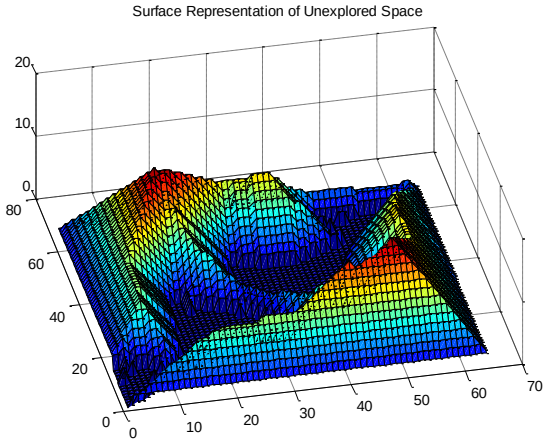

Figure 3. A surface plot of representing the Voronoi diagram. Red areas are farthest from the edge of explored space.

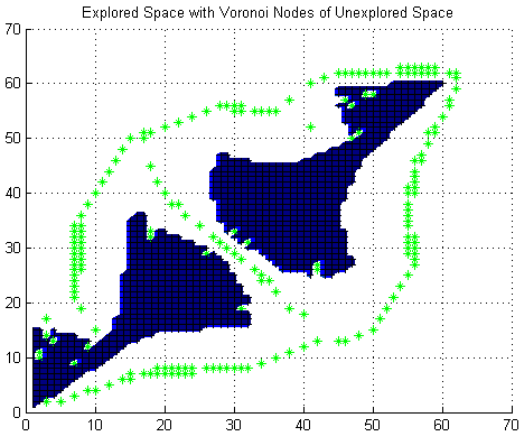

Figure 4. Overhead view of explored space with the candidate aimpoints plotted in green.

Using this information, each aircraft in the network selects an aimpoint which balances the competing objectives of maximizing new area measured with flying the minimum distance. The aircraft flies toward this point until the mapping algorithm selects another aimpoint.

Hide Location Detection

Once the entirety of the search area has been observed, the resultant map is analyzed to extract information about potential hiding places for the fugitive. The work presented here follows a three-step process borrowed from computer vision: threshold the terrain map, execute blob detection, and fit a rectangle to each blob to approximate the footprint of the building.

The thresholding step used here is very simple—any pixels of the map that are higher than an arbitrary value are marked as '1', while all other pixels are marked '0'. This threshold value is selected to be the minimum characteristic height of likely hiding places. Hiding places must be high enough to exclude small changes in terrain height or changes in the height map due to bushes or small trees. Hiding places also must be high enough to ensure that individual terrain features are distinguishable from one another—occasionally, noise or aliasing in the map can cause unconnected features to appear connected.

The algorithm then uses a single-pass blob detection routine (Ref. 17) to identify distinct structures in the search area. Blobs that are only a single pixel are excluded as they are unlikely to be large enough to represent a real hiding place.

The individual blobs are then fit to rectangular shapes using a moment of inertia analysis. The total "inertia" about the north and east axes is computed along with the "center of mass". The moments and products of inertia are computed at the blob center using the parallel axis

theorem, and finally the eigenvectors of the inertia matrix are computed to find the principal axes of each blob. Since it is assumed that the hiding areas are rectangular buildings, the eigenvalues can be used to compute the estimated length and width of each. Figure 5 shows the result from an example case of this process.

Figure 5. Output of an example building-detection routine.

Data Sharing

A key part of collaboration is effective and efficient communication. This work continues to build on the work presented in Reference 15. For this scenario, the aircraft must provide the following data: observations of terrain, current mode of operation, , and own-ship position and velocity. Adding to the challenge is that we want each aircraft to operate relatively independently, yet maintain a nearly common operating picture. Finally, the aircraft need to send only essential information at some minimum rate in order to preserve limited wireless bandwidth.

Here we assume that each aircraft trusts own-ship sensors over all others. This is a reasonable assumption for shared terrain data, as terrain sensors on the own-ship are subject to fewer errors and are likely to be providing more relevant information. Second, we assume that recent observations are more trustworthy than older observations.

When the aircraft receives a terrain map update from the others in the network, it accepts the other aircraft data unless the data conflicts with own-ship data *and* the own-ship data is younger than a specified age.

Once the aircraft transition from mapping mode into search mode, one of the aircraft (selected arbitrarily) runs the building detection routine and sends the list of buildings to the other. The aircraft from that point forward share changes to their mode and the number of the building which they will search next.

Finally, the aircraft share their state data in order to facilitate collision avoidance. This data is a trivial amount of information compared to the terrain map updates.

Collision and Obstacle Avoidance

An implied task during multi-ship operations close to terrain is to avoid collision with both obstacles and other aircraft. This work relies on previously reported results for obstacle avoidance (Ref. 14), which was accomplished by commanding a high-performance climb to maintain a specified clearance from terrain or obstacles laterally and vertically. Collision with other aircraft is avoided by cancelling the own-ship's commanded velocity component toward an other-ship when within a specified clearance distance.

Vision-based Mapping

The collaborative mapping and search system described in this paper is designed to be agnostic to the vehicle or sensing method. The searching vehicle need only be capable of generating location data for the map and/or following the commanded trajectory during search. This allows heterogeneous agents to participate in the collaborative mission. One particularly attractive method for generating location data for the map is to use a monocular camera. Monocular cameras are inexpensive and are standard payloads on many UAVs. This section outlines the monocular vision-based mapping system integrated into collaborative mapping and search. More details can be found in Reference 18.

The monocular vision-based mapping system uses feature points from a video stream to initialize and update a database of 3D point locations and associated covariances. The database of point location forms a state vector for an extended Kalman filter. As new video frames become available, image features are extracted and matched to the expected image location of database points. The resulting residual is used to update the state vector and covariances of the database points. Points that converge to within a specified uncertainty are used to update the evidence grid terrain model.

The 3D point locations in the database are parameterized as presented in Ref. 19 with an anchor point in space (in this work the location of the vehicle when the point was initialized) and the azimuth, elevation and inverse depth to the feature location. The use of this parameterization has a key advantage over a Cartesian parameterization: the uncertainty distribution of the feature points is more nearly Gaussian in inverse depth, and therefore better suited to the the EKF estimation framework. This fact results in faster convergence in

depth, a crucial requirement for fast searching and using the terrain map for obstacle avoidance.

Figures 6 and 7 show a simulation of the vision-based mapping system integrated with the obstacle avoidance system. The vehicle was flown in an oval and an obstacle was placed in the middle of one of the legs. The path of the vehicle is shown in Figure 6, and the resulting terrain map is shown in Figure 7.

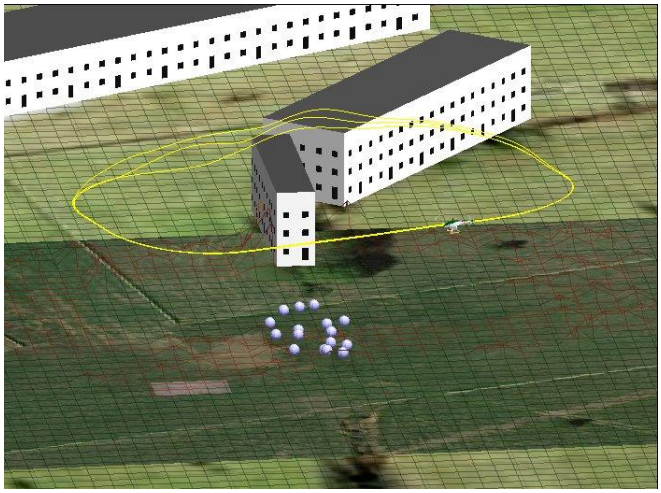

Figure 6. Flight path used for vision-based mapping.

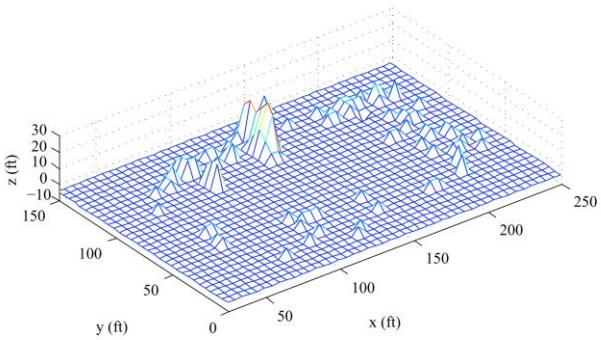

Figure 7. Example terrain map produced by monocular camera feature points.

Search Sequencing

The process to select the next search location is a "greedy-with-exceptions" heuristic. While it is not necessarily optimal with time, it is straightforward, is consistent with the level of decentralization in the system, is light on resource usage, and unpredictable.

The aircraft considers the nearest unsearched hide location, and checks if that location is legitimately claimed by another aircraft. If another aircraft is closer and would also select the location as its next destination, then the candidate is rejected and a new candidate is considered. Otherwise the candidate is selected and the aircraft transitions to search mode. For the purposes of this work, "searching" a hide area, means simply flying around the perimeter of the search area.

TEST AIRCRAFT AND EXPERIMENT

GTMax

A pair of Yamaha RMAX-based research UAVs, Figure 8, were utilized for the simulation and flight test activities under this effort (Ref. 20). The system consists of four major elements: the basic Yamaha RMAX airframe, a modular avionics system, baseline software, and a set of simulation tools.

Figure 8. Yamaha RMAX instrumented with Differential GPS, inertial measurement, magnetometer, sonar altimeter, and camera.

Figure 9. The second RMAX with nose-mounted laser range finder.

The hardware components that make up the baseline flight avionics include general purpose processing capabilities and sensing. The research avionics configuration includes:

- 2 Embedded PCs
- Inertial Sciences ISIS-IMU Inertial Measurement Unit
- NovAtel OEM-4, differential GPS

- Prosilica GC1380H monocular camera
- Sick LD-MRS laser scanner, Figure 9
- Custom made ultra-sonic sonar altimeter
- Honeywell HMR-2300, 3-Axis magnetometer
- Actuator control interface
- Vehicle telemetry (RPM, Voltage, Remote Pilot Inputs, low fuel warning)
- 11 Mbps Ethernet data link and an Ethernet switch
- FreeWave 900MHz serial data link

The baseline navigation system running on the primary flight computer is a 17 state extended Kalman filter. The states include: vehicle position, velocity, attitude (quaternion), accelerometer biases, gyro biases, and terrain height error. The system is all-attitude capable and updates at 100 Hz (Ref. 21). The baseline flight controller is an adaptive neural network trajectory following controller with 18 neural network inputs, 5 hidden layer neurons, and 7 outputs for each of the 7 degrees of freedom (Ref. 22). These 7 degrees of freedom include the usual 6 rigid-body degrees of freedom plus a degree of freedom for rotor RPM. The baseline flight controller and navigation system, which coupled with the simple baseline trajectory generator, is capable of automatic takeoff, landing, hover, forward flight up to the maximum attainable by the helicopter (around 85 feet/sec) and aggressive maneuvering.

Test Parameters and Metrics

To test the map and search architecture, the aircraft were presented with a search area modeled on McKenna MOUT site at Fort Benning, Georgia, and initially stationed outside of it with sensors oriented away from the area. The aircraft were then commanded to execute the search, with the experiment repeated varying commanded search speed and one versus two aircraft involved (see Table 1).

Table 1. Experiment Parameters.

Parameter	Value
Nominal Search Speed	5 ft/s to 30 ft/s, increments of 5 (single ship)
Max. Acceleration	5 ft/s^2
Mapping Altitude	75 ft
Sensor Range	~200 ft
Map Resolution	10 ft
Threshold Height	10 ft
Plan Max Update Rate	1 Hz
Search Area Dimensions	480 ft x 530 ft
Commanded Aircraft Separation	75 ft (~6 rotor diameters)

To compare the performance of the different search parameters, we measured the time to complete the mapping task. For the multi-aircraft scenario, the distance between aircraft and the amount of terrain updated per transmission were also reported. Typical flight paths over terrain will also be presented.

SIMULATION RESULTS

Single-Aircraft

Figure 10 shows a typical flight path history overlaid on imagery of the simulated search area. The starting position of the aircraft is denoted by the small green square, the final position by the small red triangle. The cyan rectangle denotes the limits of the search area. Note that the aircraft will travel outside the search area if it positions the sensor in a way to view the desired aim point. Other simulations with the same parameters resulted in similar though not identical flight paths. Though this pattern is clearly the fastest, the randomness will make the UAV a much harder target to either predict, engage, or evade.

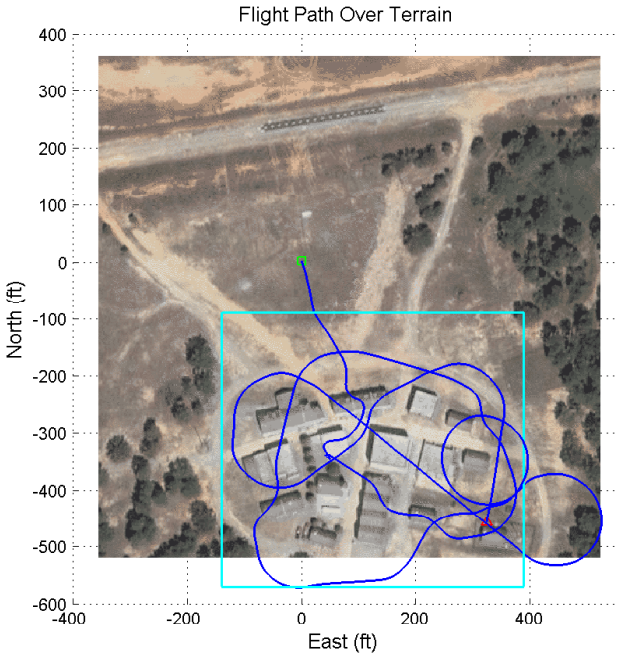

Figure 10. Flight path over search area, single aircraft at 20 ft/s.

Figure 11 shows a comparison of the mapping times for the several commanded mapping speeds. The relationship between the two seems to be roughly quadratic. Low flight speeds simply cover the ground at a very slow pace. At higher flight speeds, however, the aircraft may not have enough control authority to point its sensor at the desired location. For this aircraft and these parameters, it appears the optimal search speed is between 15 and 20 ft/s.

The building detection algorithm accurately identified the set of buildings in every scenario, with results each time similar to those shown in Figure 5.

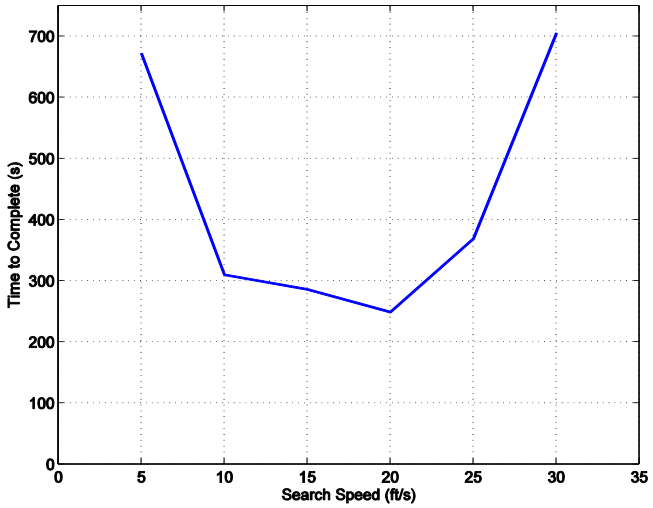

Figure 11. Time to complete mapping at varying mapping speed.

Multiple Aircraft

Introducing a second aircraft helped speed the accomplishment of the task, though not drastically. A flight path for a mapping speed of 5 ft/s is shown in Figure 12. Again, the flight paths are quite random-looking, though due to the much slower flight speeds are far more angular. Again, the building detection logic provided accurate results in each case. The task completion times were significantly improved at the high and low ends of the simulated speeds, and more modestly at the mid-range.

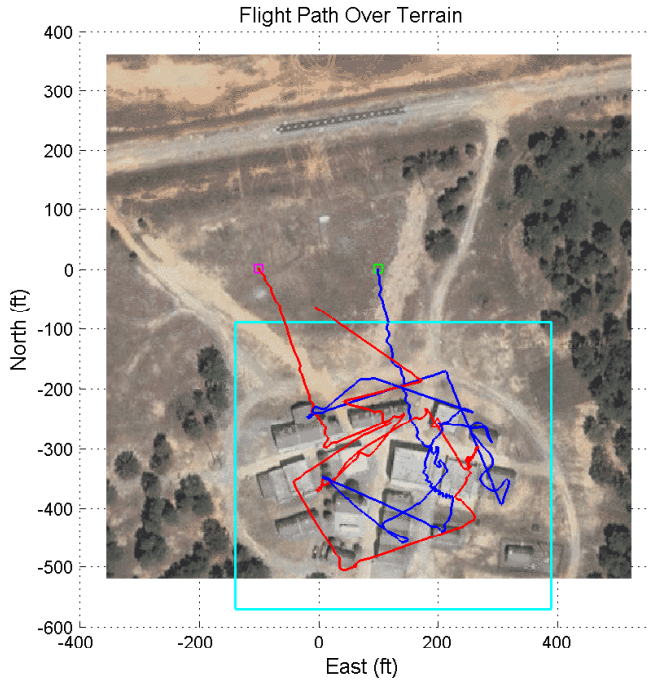

Figure 12. Flight path over search area, two aircraft at 5 ft/s.

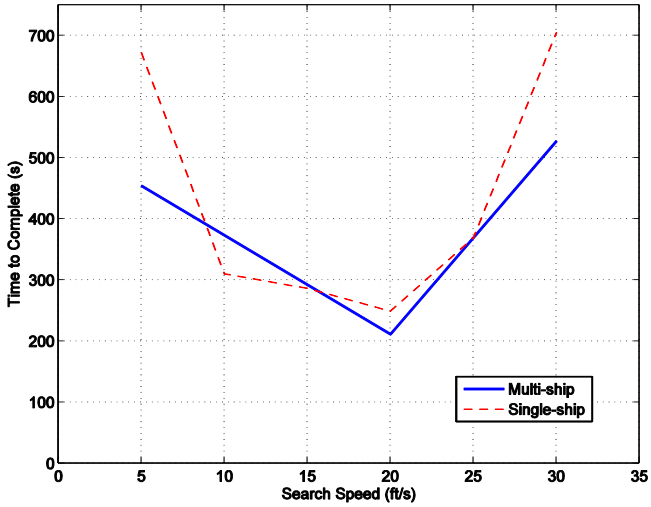

Figure 13. Time to complete mapping at varying mapping speeds, comparison between single and multi-aircraft scenarios.

This result suggests that there an optimal configuration can be found in general for a given search area, sensor capability, and set of available aircraft. If such a relationship can be found, this search strategy could be readily scaled up or down for other applications. Additionally, the example search area used in this work was relatively dense with buildings, which limited the sensors' instantaneous range as both work on line-of-sight. A more spread out or flat area would probably be completed in significantly less time.

Figure 14 shows the aircraft separation over the course of the task, as compared to the minimum commanded separation distance. This periodicity of this graph suggests that at higher speeds, the aircraft seem to get into phase with each other—flying in toward an aim point at the same time, then circling outward to get in position to measure the next selected point, and repeating. The graph also confirms that the simple collision avoidance logic is effective.

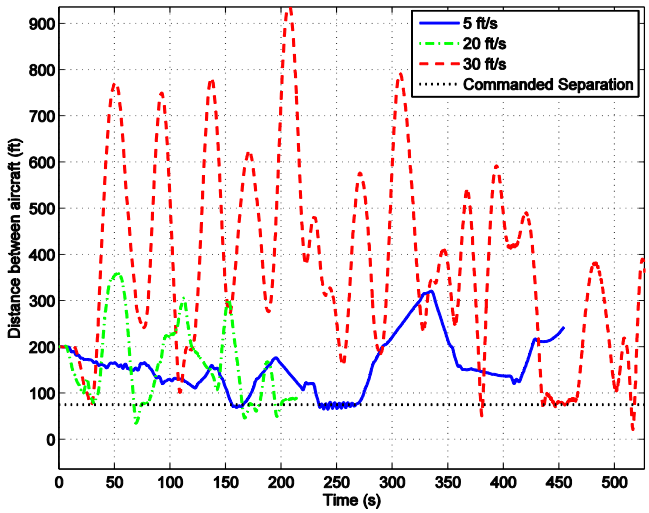

Figure 14. Aircraft separation, multiple flight conditions.

CONCLUSIONS

A mapping-and-search architecture was demonstrated single and multiple aircraft with heterogeneous sensor packages at varying flight speeds. A roughly quadratic relationship between the mapping speed and task completion time was shown. With the addition of a second aircraft, performance was improved, particularly in the regimes where the single-ship performed poorest. Future work includes further refinement and multi-aircraft flight test to validate simulation results. Additionally, the results suggest future investigation to find the relationship between the parameters and performance and to find a scale-independent measure of a search task. Finally, further work is required to address aspects of the challenge problem not considered this work, such as suspect identification, tracking, and collaborative shepherding.

ACKNOWLEDGMENTS

The authors would like to acknowledge support from Sikorsky Aircraft Corporation and the Georgia Institute of Technology, and the help of other contributors including: Jeong Hur, Henrik Christopherson, Stephen Haviland, Fritz Langford, Gerardo Delatorre, Dmitry Bershadsky, and Claus Christmann.

REFERENCES

[1]Arkin, R., Balch, T., "Behavior-based formation control for multirobot teams." *IEEE Transactions on Robotics and Automation*, Vol. 14, (6), Dec. 1998. Pp 926-939. doi: 10.1109/70.736776.

[2]Balch, T., "Social Potentials for Scalable Multi-Robot Formations." IEEE International Conference on Robotics and Automation, San Francisco, CA, April 2000.

[3]Parker, L., "Heterogeneous multi-robot cooperation." AI-TR 1465, Feb 1994.

[4]Bruce, J., Veloso, M., "Real-time randomized path planning for robot navigation," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.

[5]van Den Berg, Jur, et al, "Centralized path planning for multiple robots: Optimal decoupling into sequential plans," *Robotics: Science and systems V* 2, 2009.

[6]Luna, R., Kostas E. B., "Efficient and complete centralized multi-robot path planning," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.

[7]Godbole, D.N, "Control and coordination in uninhabited combat air vehicles," Proceedings of the American Control Conference, Jun 1999.

[8]Ribichini, G., and Frazzoli, E., "Efficient coordination of multiple-aircraft systems." Proceedings. 42nd IEEE Conference on Decision and Control, 2003.

[9]Anderson, B., et al. "UAV formation control: Theory and application," *Recent Advances in Learning and Control*, 15-33, 2008.

[10]Macswan, A., Nejat, G., Benhabib, B., "Optimal Deployment of Robotic Teams for Autonomous Wilderness Search and Rescue," 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, September 2011.

[11]Yoon, S., and Qiao, C., "A New Search Algorithm using Autonomous and Cooperative Multiple Sensor Nodes," 26th IEEE International Conference on Computer Communications, May 2007.

[12]Weiss, S., Scaramuzza, D. and Siegwart, R., "Monocular SLAM-based Navigation for Autonomous Micro Helicopters in GPS-denied Environments," Journal of Field Robotics, Vol. 28, No. 6, pp. 854-874, 2011.

[13]Chawdhary, G., et al., "GPS-denied Indoor and Outdoor Monocular Vision-aided Navigation and Control of Unmanned Aircraft," Journal of Field Robotics, Accepted, 2013.

[14]Johnson, E., et al, "Flight Testing of Nap-of-the-Earth Unmanned Helicopter Systems," American Helicopter Society 67th Annual Forum Proceedings, Virginia Beach, VA, May 2011.

[15]Johnson, E., et al, "Terrain Height Evidence Sharing for Collaborative Autonomous Rotorcraft Operation," 5th International Specialists' Meeting on Unmanned Rotorcraft and Network Centric Operations, Scottsdale, Arizona, January 22-24, 2013.

[16]Scherer, S., et al, "Flying Low and Fast Among Obstacles: Methodology and Experiments," *The International Journal of Robotics Research*, Vol. 27, No. 5, pp. 549-574, 2008.

[17]"Connected Component Labeling," Wikipedia, http://en.wikipedia.org/wiki/Connected-component_labeling

[18]Johnson, E., Magree, D., "Monocular Visual Obstacle Avoidance For UAVs," The 2013 International Conference on Unmanned Aircraft Systems, Atlanta, Georgia, May 2013 (submitted).

[19]Civera, J., Davison, A.J., Montiel, J., "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, Vol. 24, (5), pp. 932-945, 2008.

[20]Johnson, E.N. and Schrage, D.P., "System Integration and Operation of a Research Unmanned Aerial Vehicle," AIAA Journal of Aerospace Computing, Information, and Communication, Vol. 1, No. 1, pp. 5-18, January 2004.

[21]Dittrich, J.S. and Johnson, E.N., "Multi-Sensor Navigation System for an Autonomous Helicopter," Proceedings of the 21st Digital Avionics Systems Conference, October 2002.

[22]Johnson, E.N. and Kannan, S.K., "Adaptive Trajectory Control for Autonomous Helicopters," AIAA Journal of Guidance, Control, and Dynamics, Vol. 28, No. 3, pp. 524-538, May/June 2005.